

The Impact of Technical Domain Expertise on Search Behavior and Task Outcome

Julia Kiseleva¹

Alejandro Montes García¹

Jaap Kamps²

Nikita Spirin³

¹Eindhoven University of Technology, Eindhoven, The Netherlands
{j.kiseleva,a.montes.garcia}@tue.nl

²University of Amsterdam, Amsterdam, The Netherlands
kamps@uva.nl

³University of Illinois at Urbana-Champaign, Urbana, IL
spirin2@illinois.edu

ABSTRACT

Domain expertise is regarded as one of the key factors impacting search success: experts are known to write more effective queries, to select the right results on the result page, and to find answers satisfying their information needs. Search transaction logs play the crucial role in the result ranking. Yet despite the variety in expertise levels of users, all prior interactions are treated alike, suggesting that weighting in expertise can improve the ranking for informational tasks. The main aim of this paper is to investigate the impact of high levels of technical domain expertise on both search behavior and task outcome. We conduct an online user study with searchers proficient in programming languages. We focus on Java and Javascript, yet we believe that our study and results are applicable for other expertise-sensitive search tasks. The main findings are three-fold: First, we constructed expertise tests that effectively measure technical domain expertise and correlate well with the self-reported expertise. Second, we showed that there is a clear position bias, but technical domain experts were less affected by position bias. Third, we found that general expertise helped finding the correct answers, but the domain experts were more successful as they managed to detect better answers. Our work is using explicit tests to determine user expertise levels, which is an important step toward fully automatic detection of expertise levels based on interaction behavior. A deeper understanding of the impact of expertise on search behavior and task outcome can enable more effective use of expert behavior in search logs—essentially make everyone search as an expert.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Query formulation, Search process, Selection process*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Workshop on Query Understanding and Reformulation for Mobile and Web Search (QRUMS) '16, San Francisco, California, USA
Copyright 20XX ACM X-XXXXX-XX-XX/XX ...\$15.00.

Keywords

Expertise, Search behavior, Task success

1. INTRODUCTION

Users exhibit remarkably different search behavior, due to various differences including domain expertise that can greatly influence their ability to carry out successful searches. The broad motivation of our work is to investigate how we can make non-experts search like experts. Users' domain expertise is not the same as search expertise [11]. It concerns knowledge of the topic of the information need and it does not regard knowledge of the search process. So far the differences in search behavior between experts and non-experts have been examined from the perspective of: (1) a query construction process; (2) search strategies; (3) and search outcomes. In early work, Marchionini et al. [9] found that general experts in computer science, business/economics, and law, searched more content driven, and used more technical query terms. Using massive logs, White et al. [11] observed differences in source selection (hostnames, TLDs), engagement (click-through rates), and vocabulary usage, for users with general expertise in medicine, finance, law, and computer science. In a user study, Cole et al. [4] show that task and domain knowledge are beneficial to link selection in medical literature search. In related work, Zhang et al. [13] shows that we can predict the medical domain expertise level based on the variables in a user study.

In this paper, we focus on high levels of technical domain expertise, specifically experts proficient in two programming languages: Java and JavaScript, and are interested in the distinction with general programming expertise. This is different from earlier work comparing general experts against non-experts, as we focus on comparing degrees of technical domain expertise amongst experts, trying to find out if high levels of proficiency make a difference. We study interactions with the Search Engine Result Page (SERP) by users with advanced technical domain expertise. The scenario of users' interactions with a SERP is simple: a user runs a query Q and the search engine retrieves the results ranked based on a relevance score [5]. We want to understand if technical domain experts are able to find 'better' answers to their queries by exploring the SERP. The study is motivated by our own frustration when using web search engines and Q&A sites for technical how to questions, where the top ranked answers often are not the best results.

Evaluation of relevance ranking has traditionally relied on explicit human judgments or editorial labels. However, human judgments are expensive and difficult to derive from a broad audience. Moreover, it is difficult to simulate realistic information needs. User clicks are known as a good approximation towards obtaining *implicit user feedback* [2]. This relies on the basic assumption that users click on relevant results. Therefore, leveraging click-through data has become a popular approach for evaluating and optimizing information retrieval systems [3]. However, it is well known that user behavior on the SERP is biased. Different types of biases are discovered including: (1) a *position bias* [7, 8], users have a tendency to click on the first positions; (2) a *snippet bias* [12], snippets influence user decisions; (3) a *domain bias* [6], that shows that users are already familiar with Internet and they are influenced by the domain of the URL; (4) a *beliefs bias* [10], users beliefs affect their search behaviour and their decision making. Taking into consideration these biases suggests that not all clicks are equally useful for optimizing a ranking function. For one thing only successful clicks (satisfied or SAT clicks) should be taken into consideration. Our expectation is that users with a high level of proficiency are less affected by these biases.

Ageev et al. [1] proposed flexible and general informational search success model for in-depth analysis of search success which was tested based on game-like infrastructure for crowdsourcing search behaviour studies, specifically targeted towards capturing and evaluating successful search strategies on informational tasks with known intent.

The main research question studied in this paper is: *How does technical domain expertise influence search behavior?* We are particularly interested in what level of expertise is needed to make a difference: can we only trust the interactions of technical domain experts that essentially know the answers, or is a general familiarity with the domain sufficient? We conduct an user study with explicit tests to derive user expertise level, to determine the impact of technical domain expertise. This is an important step toward fully automatic detection of expertise levels based on interaction behavior. We have three concrete research questions:

RQ1: *How can we measure technical domain expertise?*

RQ2: *What is the impact of technical domain expertise on the search process?*

RQ3: *What is the impact of technical domain expertise on search outcome?*

In order to answer our research questions, we organize a user study where we are trying to imitate a realistic scenario of search tasks in a technical domain. Study participants are all having some programming background but different levels of knowledge of two programming languages: Java and JavaScript. A typical use case is that experts use search engines to re-find an answer which is common practise especially in the programming domain. For domain experts it is easy to verify if a page contains the required answer. Participants with only general domain expertise typically are proficient in one programming language, say Java, but search for information in a unfamiliar programming language, say JavaScript. Their fragmented understanding of the new language makes it much harder for them to recognize if a page contains the needed answers. We imitate this scenario in our user study.

This paper is structured as follows: §2 details the setup of our user study, while §3 discusses the results, and we conclude in §4.

2. USER STUDY DESIGN

In this section, we explain the experimental setup of the online user study.

Selecting Participants The call for participation was targeted to people interested in programming to make sure that they are likely to have information needs used in the study, using a snowball sampling approach. By doing so, we tried to make our study as realistic as possible. As an expertise field we selected two programming topics, namely, Java and JavaScript. However, users who are familiar with programming in general but not with the two selected topics also participated. We tried to keep balance between the technical domain experts and the general experts in our study.

Measuring Domain Expertise Prior to starting the study, users were asked to (1) provide some basic demographic data, namely age, gender and education level; (2) self-report their programming level in general, but also their skills in Java and JavaScript. Our study consist of two sets of tasks for each of the programming languages. The order of the topics in the study is done randomly.

Substantial research has been done in order to propose strategies to estimate users’ expertise [11]. In order to have a reliable measurement, we ask participants to fill out a questionnaire related to one of the topics. Users were not allowed to use a search engine in this step, they had to answer the questions based on their knowledge. Each questionnaire consists of ten questions. We use the results of the questionnaire in order to identify a users’ expertise level by assigning them a score $\in [0, 1]$ based on their answers.

Designing User Study The study consists of two sets of ten tasks or questions that related to the programming language. Our tasks were modeled after those that users post in specialized Q&A sites. For example, one of the tasks for Java was: “Can you override a static method in Java?” To illustrate the user study design, we provide a screenshot of one of the questions that the user had to fill, which can be seen in Figure 1.

We provide a search query for each question that is used to retrieve a SERP with ten results from the Bing API.¹ The given SERP is randomly re-ranked in order to estimate the position bias. The participants are asked to find the answer and to submit this URL. In addition, we ask participants (1) to tell us whether the query is formed in a right way or not; (2) to indicate if they knew the answer upfront. We collected ground truth answers from an expert who judged all shown results.

After finishing the study, users are asked to report again their proficiency in programming, Java and JavaScript to see if they changed their self-consideration after the test. Interestingly, five attenders have changed their mind. They could also enter open comments. For example, we have got the following comment: ‘Turns out there are some concepts which have ‘faded’ a bit in my memory!’, which basically shows that even good developers sometimes need to refresh their memory.

¹<http://datamarket.azure.com/dataset/bing/search>

Question

Can you override a static method in Java?

Question 1 out of 10

Answer

Enter your Answer

Enter here the answer to the question

Selection Source

Source of the answer

- ✓ The answer cannot be found in the URLs provided
- 1: <http://stackoverflow.com/questions/2475259/can-i-override-and-overload-static-methods-in-java>
- 2: <http://java67.blogspot.com/2012/08/can-we-overload-static-method-in-java.html>
- 3: <http://stackoverflow.com/questions/2223386/why-doesnt-java-allow-overriding-of-static-methods>

Is the query "override static method Java" good enough to answer the question?

✓ Yes
No

Query Satisfaction

Did you know the answer without needing the URLs provided?

✓ Yes
No

Knowledge about question

Click to submit answer

Submit

List of URLs

Query: override static method Java

Query and Randomized SERP

1: Can I override and overload static methods in Java ...
I'd like to know: Why can't static methods be overridden in Java? Can static methods be overloaded in Java?

2: Can we overload static method in Java Program - Example ...
Yes, we can overload static method in Java. In terms of method overloading static method are just like normal methods and in order to overload static ...

Figure 1: Screenshot of the online user study.

3. RESULTS

In this section, we try to answer our three main research questions. First, we look at the collected data and the results from the domain knowledge tests. Second, we look at how the position bias depends on user expertise and how the number of correct answers depends on user expertise.

Measuring Technical Domain Expertise We now describe the collected data, and try to answer our first research question: **RQ1: How can we measure technical domain expertise?**

In total, we have 29 participants in our study. From the demographic perspective our dataset can be characterized by participants age and education level. In terms of education level we have the following population of participants: High school 8%, Bachelor 12%, Master 56%, PhD 24%. In terms of age we have the following population of participants: 18-23 years is 16%, 24-29 years is 36%, 30-35 years is 28%, 36-42 years is 16%, 43-48 years is 4%.

Figure 2 shows the distribution of expertise in Java (top half) and JavaScript (bottom half). The expertise scores are calculated based on ten questions from the pre-survey (described in §2). As we can see, the majority of the par-

ticipants in the study have high levels of expertise in either Java or JavaScript.

Figure 3 shows the relation between the participants' expertise scores in Java and JavaScript. The relation between the skills in Java and JavaScript is weak (Pearson correlation of 0.44, $p < 0.05$) signaling that only a small fraction of participants has high levels of expertise in both.

Figure 4 shows the expertise test scores over the self-reported expertise levels. We see a clear relation between the self reported expertise levels and the test scores, the correlation is 0.60 for Java (Pearson, $p < 0.01$) and 0.75 for JavaScript ($p < 0.001$). This result gives confidence in the tests to quantify the technical domain expertise of the participants.

Our main finding is that the expertise test is effective and correlates well with the self-reported expertise. This implies the usefulness of the test, but also validates the self-reported expertise score as a reliable indicator of user expertise.

Impact on Search Behavior We now investigate our second research question: **RQ2: What is the impact of technical domain expertise on the search process?**

We calculate the distribution of positions over all submitted users answers for both topics (a position of URL that

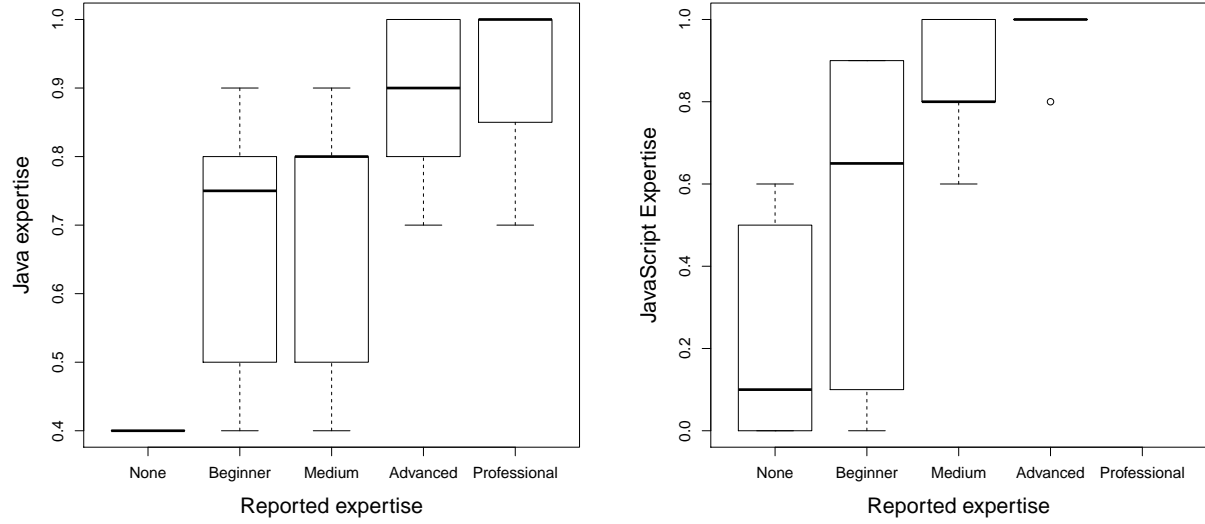


Figure 4: Expertise test results over self-reported scores for Java expertise (left) and JavaScript expertise (right).

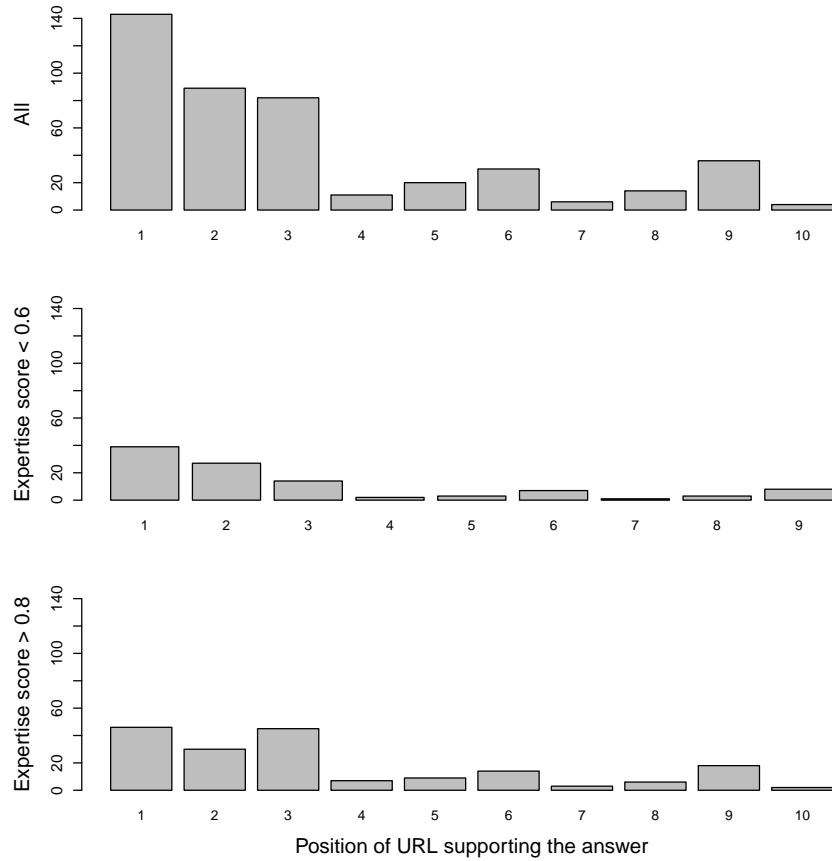


Figure 5: Position bias with respect to the users' expertise.

study participants submit as a source of the correct answer to the proposed question). This event can be mapped to the SAT click in web search behavior. Figure 5 presents the distribution of selected answers with regard to the calculated

users' expertise score: over all participants (top), over those with relatively low levels of expertise (< 0.6 , middle), and over those with a relatively high levels of expertise (> 0.8 , bottom). We performed a goodness of fit test against a uni-

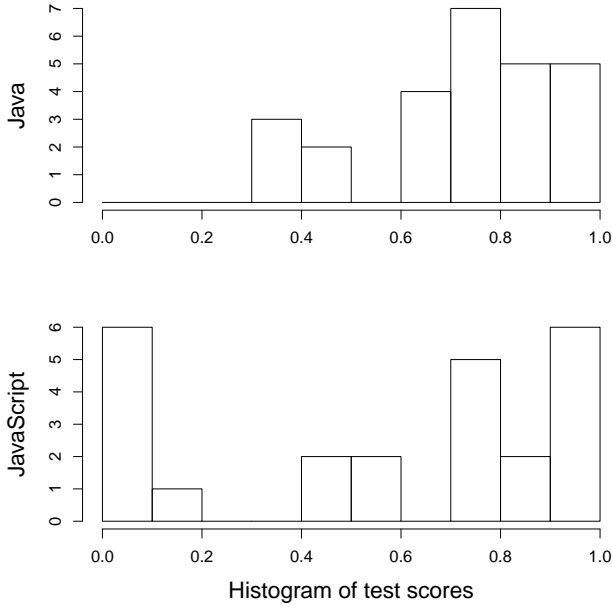


Figure 2: A histogram of expertise score of participants calculated for Java (top) and JavaScript (bottom).

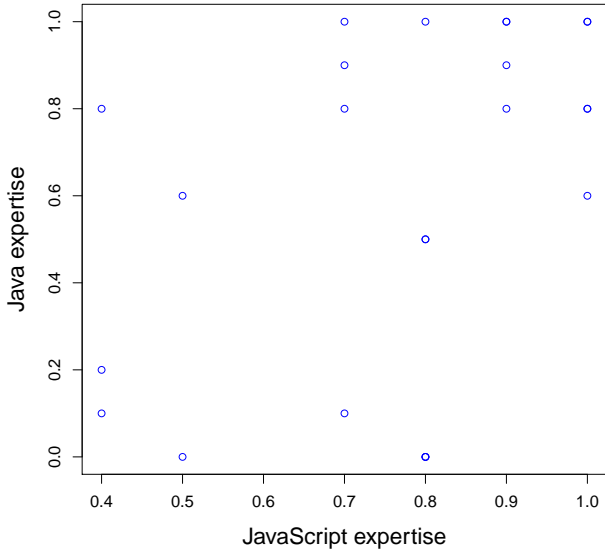


Figure 3: A scatter plot of the calculated skill score for both topics: Java and JavaScript.

form distribution which fails convincingly (χ^2 goodness of fit test, $p < 0.001$ for all three cases).

Recall that the SERPs were randomized, hence relevant results are uniformly distributed across the ten positions. We can clearly see that the whole population is biased to the position of the result on the SERP, showcasing that results are scanned from top to bottom. We see an even stronger position bias for those with lower test scores, while for those with a higher test score, the position bias is less pronounced.

The technical experts more frequently pick a lower ranked results – although even the experts clearly prefer top ranked results.

The position bias as shown in Figure 5 does not show a monotonically declining pattern, as some positions such as 6 and 9 are more popular than others. Closer inspection reveals that this is due to the popularity particular Q&A sites, in particular <http://stackoverflow.com/>, that attract attention. As we are working with a single randomized SERP for each question—allowing us to compare position across participants—the distribution of popular Q&A sites is not exactly uniform over the sample. So in addition to the position bias, we see a domain bias.

We can see an evidence of the snippet bias in participants behavior as they are selecting ‘correct’ result URLs without clicking on them. Indeed, for these cases we can see that answers are provided in snippets.

Our main finding is that the click distribution shows evidence of the position bias of all participants. However, for the experts the position bias is less pronounced and they tend to check the lower positioned results.

Impact on Search Outcome We now examine our third research question: **RQ3: What is the impact of technical domain expertise on search outcome?**

In order to collect the ground truth for correctness of the provided answers, we pooled results from the higher scoring experts and had the answers and URLs judged by an expert editorial judge. As it turned out, on average 5 out of 10 results supported answering the task, varying between 2 and 8 per question.

Figure 6 shows the distribution of correct answers over expertise levels for Java (left) and for JavaScript (right). We see a clear relation for both Java and JavaScript: higher expertise levels lead to higher fractions of correct answers. The relation is highly significant (Pearson χ^2 , $p < 0.0001$) for both Java and JavaScript. There is an interesting deviation for those scoring very low on Java, yet producing many correct answers. A plausible explanation is that these participants have sufficient passive understanding, hence can recognize answers pages on the information on the web pages, but cannot actively produce this in the test. This is supported by their relatively high fractions of “I don’t know” answers on the Java expertise test.

Our main finding is that the participants with general programming expertise are able to find out the correct answers on SERP but they tend to select higher positioned pages in SERP. Clearly, the experts manage to detect better answers as they dig them from the bottom of SERP.

4. CONCLUSIONS AND DISCUSSION

The main aim of this paper was to investigate how technical domain expertise influences search behavior, focusing on high levels of proficiency in programming languages versus general programming expertise. We studied three concrete research questions. First, we investigated: **RQ1: How can we measure technical domain expertise?** Our main finding was that the expertise test is effective and correlates well with the self-reported expertise. Second, we looked at: **RQ2: What is the impact of technical domain expertise on the search process?** Our main finding was that the distribution of SAT clicks exhibited an evidence all participants were biased the URL position on SERP. However, the technical

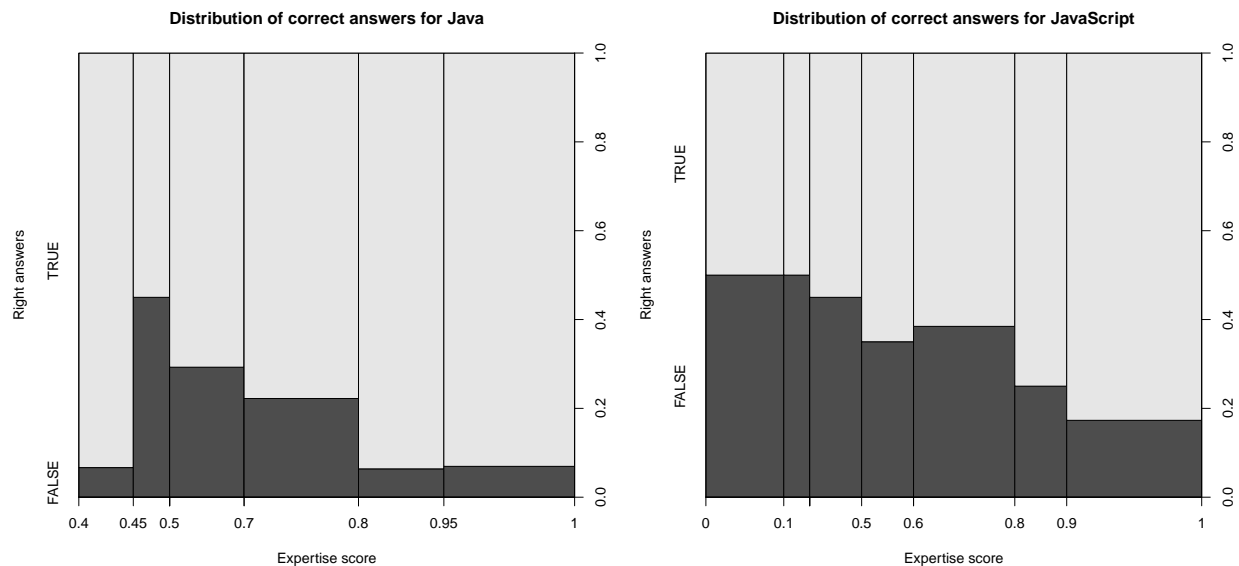


Figure 6: A distribution of right answers of participants with different expertise levels in Java (left) and JavaScript (right).

domain expert’s biases was less pronounced and they tended to check the SERP’s bottom. Third, we examined: **RQ3: What is the impact of technical domain expertise on search outcome?** Our main finding was that having a general programming expertise helped to derive the good answers on the SERP, but the experts with high proficiency managed to detect better answers as they dug them from the bottom of the SERP.

Our general conclusion is that participants with technical domain expertise behaved differently, and were more effective, than those with general expertise in the area. The differences are clear, but mostly a matter of degree, suggesting that there is value in both types of interactions. This suggests that properly weighting clicks relative to the expertise—and essentially use the expert behavior to get clicks of higher quality that hold the potential to improve the search result ranking. We are currently working on the prediction of technical domain expertise levels based on behavioral data.

Our results are on technical domain expertise, where levels of proficiency can be crisply defined, and we focused on searches related to their work task using web search engines and Q&A sites for technical how to questions. We expect our results to generalize to other specialized areas, typical of domain-specific search. In light of the earlier literature on domain expertise, these results suggest that we need to go beyond the separation of those with and without domain expertise or familiarity—the classic distinction between experts and novices—but that there is value in distinguishing high levels of domain expertise—the distinction between general expertise in the area, and those ‘who know the answer.’

Acknowledgments

The questionnaires, collected data, and the code for running the study, are available from <http://www.win.tue.nl/~mpechen/projects/capa/#Datasets>.

This research has been partly supported by STW and it is the part of the CAPA² project.

References

- [1] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: a game for modeling different types of web search success using interaction data. In *SIGIR*, 2011.
- [2] E. Agichtein, E. Brill, and S. T. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pages 19–26, 2006.
- [3] C. Brandt, T. Joachims, Y. Yue, and J. Bank. Dynamic ranked retrieval. In *WSDM*, 2011.
- [4] M. J. Cole, X. Zhang, C. Liu, N. J. Belkin, and J. Gwizdka. Knowledge effects on document selection in search results pages. In *SIGIR*, pages 1219–1220, 2011.
- [5] N. Craswell, O. Zoeter, M. J. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM*, pages 87–94, 2008.
- [6] S. Jeong, N. Mishra, E. Sadikov, and L. Zhang. Domain bias in web search. In *WSDM*, pages 55–64, 2012.
- [7] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.
- [8] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, pages 154–161, 2005.
- [9] G. Marchionini, S. Dwiggins, A. Katz, and X. Lin. Information seeking in full-text end-user-oriented search systems: The roles of domain and search expertise. *Library & Information Science Research*, 15:35–69, 1993.
- [10] R. W. White. Beliefs and biases in web search. In *SIGIR*, 2013.
- [11] R. W. White, S. T. Dumais, and J. Teevan. Characterizing the influence of domain expertise on web search behavior. In *WSDM*, pages 132–141, 2009.
- [12] Y. Yue, R. Patel, and H. Roehrig. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *WWW*, pages 1011–1018, 2010.
- [13] X. Zhang, M. Cole, and N. Belkin. Predicting users’ domain knowledge from search behaviors. In *SIGIR*, pages 1225–1226, 2011.

²www.win.tue.nl/~mpechen/projects/capa/